Global Information Assurance Certification Paper

Deception Techniques as part of Intrusion Detection Strategy

GIAC (GCIA) Gold Certification

Author: Colm Kennedy
Advisor: Mohammed Haron
Date of Acceptance: June 29th 2016

Abstract

Intrusion Detection Systems (IDS) are used to help the Security Analyst detect unauthorized or suspicious activity inside a network and on Endpoints (servers, workstations). An early stage in the Hackers methodology uses Active Recon on the network to find other machines they can pivot to and maintain their presence. With the help of deception techniques, the theory of IDS can be enhanced to see when an unauthorized connection is attempted. These deception techniques include setting up VM's that mimic actual systems on the network. These VM's are in essence a canary in a coal mine, acting as an early indicator of intrusion. This paper will dive into the effectiveness of this type of technique and how it can be used to aid in the overall IDS stance in an environment.

1. Introduction

Throughout history, deception techniques have been used successfully against adversaries to gain a tactical advantage. One instance that stands out is the Ghost Army from World War 2. Their purpose was to use deception techniques in Europe to divert the enemies' attention and reposition themselves away from where real allied troops were approaching. What made this group so successful was their ability to make their decoys look authentic to the distant eye. They went to great extents to make inflatable vehicles look authentic and used large speakers to mimic sounds of moving units on the battlefield. This tactic was successful and led to many victories of the Allied troops and saved thousands of lives as a result of their use of deception. (Beyer, 2013)

The goal of this paper is to show the effectiveness of using deception techniques to aid in the overall Intrusion Detection System in an environment. Deception can work to assist the security analyst when detecting suspicious activity on a network. Networks are the battlefront security analysts are attempting to protect and defeat any attempts to gain unauthorized access to it. Analysts need to be alerted to the presence of adversaries as early as possible if unauthorized access is occurring to that network. Intrusion Detection Systems help with this but can be bypassed. If stolen credentials are used on a network or the activity is coming from an authorized system on that network malicious activity can be missed. Deception techniques can work in conjunction with IDS to act as the canary in a coal mine.

The majority of IDS setups utilize three types of detection methods: Signature-based detection, Anomaly-based detection and Stateful Protocol Analysis. Signature Based Detections look for particular activity patterns that match a predefined signature and if there is no rule defined it will not alert. Anomaly-based Detection looks for behavior that is different from a predefined baseline of activity and alerts only if something changes from the baseline. Stateful protocol Analysis creates proper protocol state profiles. It then takes the captured activity and compares it with the profiles to determine if it's harmless or suspicious (Intrusion detection system, n.d.).

All of these have issues with common network activity being used in malicious ways, to avoid detection. Deception can be another layer that can help enhance the IDS environment

(Lateral movement, 2015). Decoy systems are deployed across a network waiting for connections to be made for alerts to fire. Deception techniques can be a layer within an IDS program to help alert on activity to these systems because they are purely there as decoys. They serve no legitimate purpose and mimic what is present on the systems and servers environment. Mimicking the production environment helps make them more appealing to an attacker attempting to scan an internal environment. With no legitimate activity expected on the decoys, almost all connections are suspicious in nature and require further investigation (Richardson, 2015).

Most attacks follow a pattern that includes the initial compromise, establishing the beachhead into the network, privilege escalation, internal recon and pivoting laterally within the network (Active Deception, 2015). These attacks are most successful when they can blend into the rest of the network traffic so not to set off any alarms. Attackers blending into an environment is where the deception decoy systems come into play. When using decoys for deception, it is best to make them look authentic. The more they mimic other legitimate systems on a network, the higher the chance an attacker will make contact with the decoy.

2. Deception Automation

In recent years there have been startup companies that help automate some deception techniques. These include "Attivo Networks Inc., Cymmetria, TrapX Security Inc. and TopSpin Security" (Richardson, 2015). Each product offers automated solutions for deception systems to be distributed across specified IP ranges. Currently, there is no industry leader as most in this field are relatively new (Richardson, 2015). For this paper, Attivo Network's product will be used.

The Attivo BOTsink Appliance will be used to show how deception tools can give visibility into common network activity used for malicious purposes. An important part of this appliance is the ability to mimic other machines on the network, making these decoys attractive targets. The appliance is able to host many VM decoy systems at once. These systems send up all activity to the appliance where events are correlated, and alerts are created based on severity of what activity is occurring. The appliance also monitors the attack while in progress to gather

intelligence on the particular attack vector that is used. Knowing the attack vector can assist a security analyst in shutting the compromise down before the attacker can extract data and possibly create rules or profiles in an IDS based on this visibility. Attivo BOTsink's concept is to have the decoy VM's placed in as many subnets across a network as possible to increase the chances of detection.

3. Proving out a Deception Tool

3.1 Lab description

This paper will focus on a small Lab environment setup using an Attivo BOTsink appliance with five VM's installed. These five systems will make up the deception framework on a small scale. The VM's consist of three Windows 7 systems and two servers - one domain controller and one file server. The VM's communicate back to the appliance where logs get correlated, and alerts are sent. Alerts can also be set to email for certain severity levels. These VM's are all controlled and run from this appliance and can be rebuilt if required. Each VM uses multiple IP addresses to cover multiple subnets. They also use randomly created unique MAC addresses assigned to each IP. Using multiple IP's across subnets allows for maximum coverage with limited VM's used. The appliance also hosts the console which controls all aspects of it and all VM's running from a central location. From the console, you can see all the IP's associated with system names and also the appropriate MAC address for each. The console is where alerts can be viewed and investigated further if necessary. All events to a VM get recorded in the console where they are correlated to determine the level of severity for an event or group of events. Alerts can also be configured for email to be sent when an event triggers a specified severity level. Their format is as follows. (Attivo Networks, 2015)

What details BOTsink Appliance alerts give us?

- Severity Level – Very High, High, Medium, Low, Very Low, Event Logged
- Timestamp
- Attacker IP
- Target OS

- Description – Type of activity
- Details – Shows data in raw form
- Category
- Service

3.2 Lab Setup

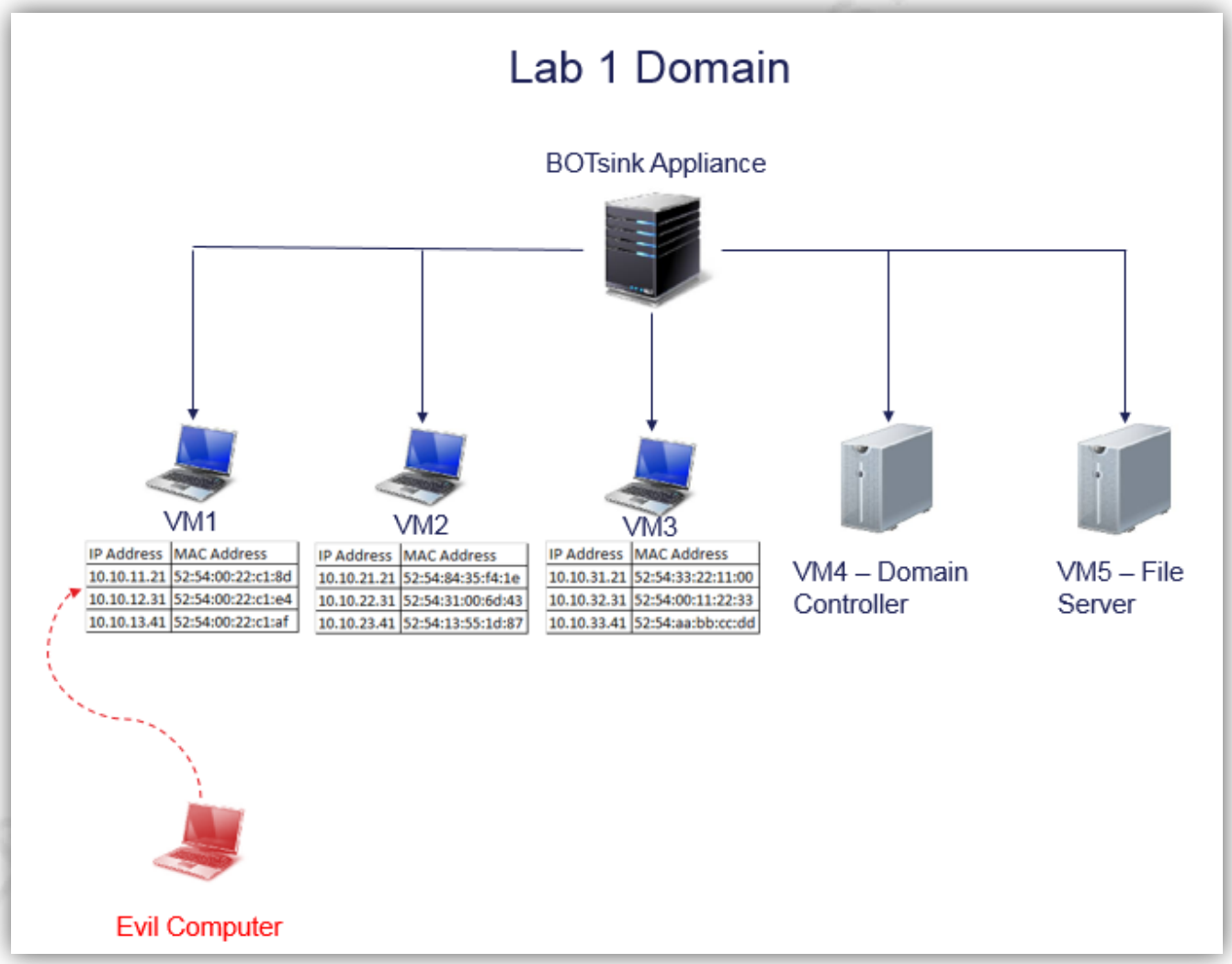

Figure 1 – Lab Setup

The basic configuration of the lab can be seen in Figure 1 that shows the BOTsink Appliance that hosts and controls the VM's. For this example, we have set up five VM's that are on this network including three Windows 7 systems, a Domain Controller and File Server (VM1-Windows 7, VM2-Windows 7, VM3-Windows 7, VM4 – Domain Controller and VM5 – File Server). Each system will pull an IP address from a respective subnet and create a fake MAC address. These systems will do this multiple times so a single VM will show up on multiple subnets with different IP addresses matching the respective subnet. Also, the VM will use a unique MAC address for each individual IP. Using multiple IP's and MAC's for a single VM allows for maximum coverage in the environment. For this lab, each system was set up to have three IP addresses and appropriate MAC addresses. The domain controllers and file servers have the same configuration. These systems are not joined to the company's production domain but are members of a separate domain. For this lab, the domain name is Lab1 Domain. In a real distribution of this tool, it is advised to use a name similar to that of the production domain name. The decoys will include innocuous traffic sent between each to simulate what other systems on the real network are doing. Traffic includes successful and failed authentications against the domain controller. The system naming convention for the VM's should match the production network. They should also use the same Operating Systems and have similar services running. Using the same naming convention and Operating Systems help with creating the authenticity requirement for the deception decoy to be effective.

For this lab, we will concentrate on VM1 and make all connections and compromise steps against it. In Figure 1 a compromised system, evil computer, is shown that signifies where the tests will originate. To save time the steps required to compromise this computer system will be skipped. The assumption for this lab is that the evil computer is a compromised system on the network that the attacker is using as a beachhead after stealing a user's credentials through social engineering attempts. With the stolen credentials the attacker now has access to a system on the network and begins internal recon to scan for other systems on the network. The internal recon is done to gain information that is valuable to an attacker to attempt lateral movement within that network. This paper will demonstrate what the attacker sees during the steps they take. Also, what the appliance is showing, how it rates the severity of each event it sees and what alerts are firing in the console.

The goal is to see what events cause the appliance to alert. To begin, a net view command will allow visibility into what other systems are visible on the network. Then a continuous ICMP Echo request will be run against one of the systems to start. As previously stated all steps for this lab will be completed against the VM1 system. Then an RDP session will be established into VM1, mimicking the act of using successfully stolen credentials. Once an RDP session is established, a new user is created and assigned to the Administrators group, then used to open an RDP session. Within the newly created users, RDP session files will be created, an attempt to copy files to the system will take place, and a service is stopped. Also, to wrap the testing up PSexec is used against VM1 and some commands will be run to see if any alerts are triggered. The key here is running basic commands that are typically done regularly by applications or system administrators on a network. The hope is that most if not all the activity that takes place will set off alerts of some severity level to make the Security team aware of activity on the decoy system.

3.3 Lab – Net View command

The first step in the lab is to see what other systems are on the network. Running the net view command will help show what systems the evil computer can see. The net view command is displayed in Figure 2. In this step, all systems are showing with their respective MAC addresses as the responses given to the net view command.

```
C:\Windows\system32>net view
Server Name            Remark

-------------------------------------
\\VM52540022c18d
\\VM52540022c1e4
\\VM52540022c1af
\\VM52548435f41e
\\VM525431006d43
\\VM525413551d87
\\VM525433221100
\\VM525400112233
\\VM5254aabbccdd
\\VM4
\\VM5
The command completed successfully.
```

Figure 2 – Net View Command and results

With the net view command, all visible systems are shown; the remaining steps will all be against the first system on the list, VM52540022c18d. As seen in Figure 1, this system is one of the three IP addresses/MAC addresses that VM1 is using. This activity did not show up in the console as any severity level or event.

3.4 Lab – ICMP Echo Request Test

When sending a continuous ICMP echo request to VM1 from the evil computer, Figure 3 shows the activity as being logged but has been categorized as a low severity level. Relatively benign behavior but could be a sign of malicious behavior or misconfigured systems/tasks on the network. Regardless it should initiate a security analyst to investigate what caused this activity from the evil computer.

Severity	Timestamp	Attacker IP	Target Host	Attack Description	Attack Details
Low	2016-02-25T12:34:45	Evil Computer	VM1	ICMP - Recon	IDS event (Attivo rule ICMP {ICMP} Evil Computer -> VM1 (VM52540022c18d) Source Domain Name (Lab1) Destination Domain Name (Lab1)

Figure 3 – ICMP alert from Console

3.5 Lab – RDP Session Test

The next test is to open an RDP session into VM1 with the administrator credentials for that system. The RDP session is seen by the appliance as an event logged but not alerted. Next, the user Gonzo is created, and that user added to the Administrators group on that system. These steps do kick off an alert in the console shown in Figure 4 and given a severity level of medium. The console shows all the actions taken in the RDP session. From the bottom up, you first see the user Gonzo being created and enabled with the password set at that time. You can see that activity gets rated at medium severity. It isn't until the user is added to the Administrators group that the high alert is triggered. At that point, the security analyst should be aware that something malicious is happening and attempt the containment of the situation.

Severity	Timestamp	Attacker IP	Target Host	Attack Description	Attack Details
High	2016-03-03T23:27:30	Evil Computer	VM1	"Administrators Group Changed"	"WinEvtLog; WinEvtLog: Security: AUDIT_SUCCESS(4732): Microsoft-Windows-Security-Auditing: (no user): no domain: VM1: A member was added to a security-enabled local group. Subject: Security ID: S-1-5-21-500 Account Name: Administrator Account Domain: VM1 Logon ID: 0x26d16ec Member: Security ID: S-1-5-21-1001 Account Name: - Group: Security ID: S-1-5-32-544 Group Name: Administrators Group Domain: Builtin Additional Information: Privileges: - "
Medium	2016-03-03T23:26:50	Evil Computer	VM1	"User account changed"	"WinEvtLog; WinEvtLog: Security: AUDIT_SUCCESS(4738): Microsoft-Windows-Security-Auditing: (no user): no domain: VM1: A user account was changed. Subject: Security ID: S-1-5-21-500 Account Name: Administrator Account Domain: VM1 Logon ID: 0x26d16ec Target Account: Security ID: S-1-5-21-1001 Account Name: Gonzo Account Domain: VM1 Changed Attributes: SAM Account Name: Gonzo Display Name: Gonzo The Great User Principal Name: - Home Directory: %%1793 Home Drive: %%1793 Script Path: %%1793 Profile Path: %%1793 User Workstations: %%1793 Password Last Set: 03/03/2016 11:26:50 PM Account Expires: %%1794 Primary Group ID: 513 AllowedToDelegateTo: Old UAC Value: 0x10 New UAC Value: 0x210 User Account Control: %%2089 User Parameters: %%1793 SID History: - Logon Hours: %%1797 Additional Information: Privileges: - "
Medium	2016-03-03T23:26:00	Evil Computer	VM1	"User account enabled or created"	"WinEvtLog; WinEvtLog: Security: AUDIT_SUCCESS(4722): Microsoft-Windows-Security-Auditing: (no user): no domain: VM1: A user account was enabled. Subject: Security ID: S-1-5-21-500 Account Name: Administrator Account Domain: VM1 Logon ID: 0x26d16ec Target Account: Security ID: S-1-5-21-1001 Account Name: Gonzo Account Domain: VM1 "
Medium	2016-03-03T23:25:30	Evil Computer	VM1	"User account changed"	"WinEvtLog; WinEvtLog: Security: AUDIT_SUCCESS(4738): Microsoft-Windows-Security-Auditing: (no user): no domain: VM1: A user account was changed. Subject: Security ID: S-1-5-21-500 Account Name: Administrator Account Domain: VM1 Logon ID: 0x26d16ec Target Account: Security ID: S-1-5-21-1001 Account Name: Gonzo Account Domain: VM1 Changed Attributes: SAM Account Name: Gonzo Display Name: Gonzo The Great User Principal Name: - Home Directory: %%1793 Home Drive: %%1793 Script Path: %%1793 Profile Path: %%1793 User Workstations: %%1793 Password Last Set: %%1794 Account Expires: %%1794 Primary Group ID: 513 AllowedToDelegateTo: - Old UAC Value 0x10 New UAC Value: 0x10 User Account Control: - User Parameters: %%1793 SID History: - Logon Hours: %%1797 Additional Information Privileges: - "

Figure 4 – RDP Session Alerts from Console

With the newly created user Gonzo, an RDP session is opened on VM1. Folders are created in this session, then a file named "EvilCode.bat" is created as shown in Figure 5, and later this file is deleted. In the console, these actions were recorded as events logged and did not reach a high enough severity to warrant an alert. The assumption is that the first high alert should be adequate to initiate further investigation into the logged activity after the fact to see what transpired during the sessions.

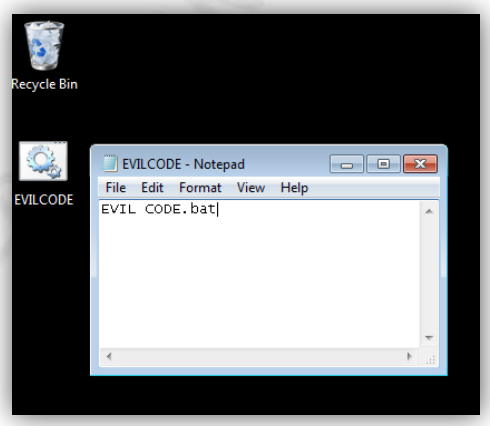

Figure 5 – Creating file EvilCode.bat

Colm Kennedy

While the user Gonzo is still logged in, an attempt to gain internet access was denied due to this functionality being disabled on these VMs. The assumption is this is done to ensure that these systems are not used maliciously against systems outside a network, reducing the residual responsibility that could arise. Disabling this also makes it difficult for an attacker to download files from external locations to these VM's. The internet denial would be the first sign to the attacker that they have stumbled upon a possible trap system. As previously attempted actions were permitted while the console was alerting in the background.

3.6 – PsExec Tool Test

Using the PsExec tool an attempt to connect with VM1 with no credentials is denied. The console records this event as a failed login attempt from the evil computer, but its severity level is set at event logged. A PsExec session with credentials is then attempted using the below command.

Psexec \\10.10.11.21 –u Gonzo –p Gonzo –s cmd.exe (Russinovich, 2014)

This command gives a command line of VM1 with Administrator level permissions, created when the user Gonzo was added to the Administrators group previously. To confirm this command line is for VM1, a simple ipconfig command is run and shown in Figure 6. It shows the successful connection, using PsExec, to the intended IP and MAC address of VM1.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ckennedy\Desktop\PsTools>psexec \\10.10.11.21 -u Gonzo -p Gonzo -s cmd.exe

PsExec v1.98 - Execute processes remotely
Copyright (C) 2001-2010 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : VM1
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : Lab1

Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . : Lab1
   Description . . . . . . . . . . . : Realtek RTL8139C+ Fast Ethernet NIC #2
   Physical Address. . . . . . . . . : 52-54-00-22-c1-8d
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.10.11.21(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Lease Obtained. . . . . . . . . . : Friday,April 15 , 2016 3:02:50 PM
   Lease Expires . . . . . . . . . . : Sunday, April 24, 2016 3:32:50 PM
   Default Gateway . . . . . . . . . : 10.10.11.1
   DHCP Server . . . . . . . . . . . : 10.10.20.20
   DNS Servers . . . . . . . . . . . : 10.10.20.20

   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Figure 6 – PsExec session with Ipconfig

While still in the PsExec session a few other commands are run below in Figure 7. The first, of which was to browse to the user's directory and view what users are currently on this system. The next step was to attempt to stop a service. As shown in Figure 7 the browser service was stopped. The browser service was chosen so not to cause the system to blue screen, allowing further testing to continue. Stopping this service demonstrates the ability to stop any service. As a surprise, this showed as a low severity event. The assumption is that due to the previous alerts from the creation of the user Gonzo all other events are logged but not necessarily marked as high severity. Also, the PsExec session was logged as authenticated and showed in the console similar to the RDP session with the Gonzo user.

In the last portion of Figure 7, the "Netstat" command was run to show all the active connections VM1 currently has with associated ports. The address that is showing in the foreign address column is the compromised system evil computer.

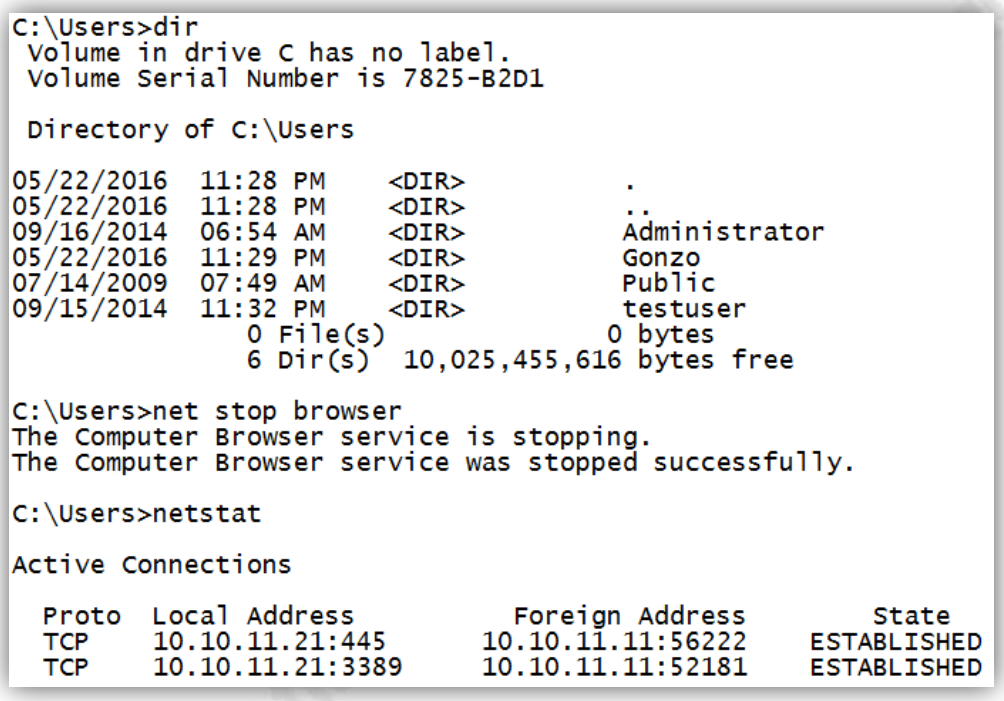

Figure 7 – PsExec Session with User Directory, Service Stop, and Netstat

As a last test calc.exe was run from the remote command line which in turn opened the calculator application on VM1. Running this command simulates opening a malicious executable on the system and running it to see if any alerts were triggered. The initial PsExec connection alerted in the console as a medium severity and showed the user Gonzo as initiating the connection. The rest of the activity run through the PsExec session on VM1 only showed in the console with low severity. The hope would be that a security analyst would have seen the initial alert and started monitoring the activities of the attacker. To confirm this is the action the console would take, follow up sessions were completed for each step in this lab. Each follow-up session

with the system, using the same procedure and commands, resulted in the console showing the same severity levels.

4. Drawbacks

As the tests proceeded through the different steps, certain activities did not alert as expected. It started with the ICMP traffic where the first alert only came in a few minutes after the continuous ping started sending traffic. With the RDP sessions, it seemed odd that creating files, running files and deleting files did not get logged at all. Yes, the initial RDP session as administrator was logged as an event, but it didn't alert in any severity. The concern is that without the initial alert a security analyst may not see these actions on decoy systems until it is too late. Similarly, with the PsExec tool, it seems that the appliance was showing that activity as low severity.

With the other commands that were run the trend was to not alert but just log the events after an initial alert fired. The lack of alerts appeared to be a result of the appliance typically not reporting immediately as it is waiting to see what other actions are happening that collectively would be marked as suspicious. It would then alert or raise the severity level as more suspicious actions are taken. Waiting for all activity before raising the severity level helps reduce the chance of false positives. However, it is a drawback because initial thoughts are that all connections to these VM's should alert unless expected traffic is whitelisted.

It makes sense to try avoiding false positives with other security tools like an IDS or anti-virus in fear of creating alert fatigue. But in the case of tools like this, there is no such thing as a false positive once setup and fine tuning is completed. Nothing should be making connections with the decoy systems as they serve no legitimate purpose.

5. Conclusion

Through these lab tests most if not all traffic to these VM's is logged or alerted on. There are some relatively simple fine tuning involved by selecting the minimum severity level setting to the desired level to be alerted. Realistically, false positives are not a bad thing as long as they don't create the dreaded alert fatigue and cause a security analyst to miss concerning activity.

These labs also show the positive way this tool can be used to observe events and better understand how the attacker is operating in your environment. Using events from the BOTsink appliance can assist the security analyst in searching for other compromised systems in the network. By tracking the attacker's actions signatures can be created, patterns of behavior or even credentials used can be flagged to help eradicate the attacker from the network. (Richardson, 2015)

Deception techniques can work in most environments because attackers typically believe what they see to be true (Attivo Networks, 2015). Using deception to misdirect them to the decoy systems, it makes it difficult for them to proceed with their compromise. Most attackers want the easy target and if frustrated with deception systems they may give up and move on to an easier target. Also, with these decoy systems, the attacker may not be able to stay in a network for very long before being discovered. These decoy systems can be beneficial in the fight against these compromises. In particular, the Attivo BOTsink Appliance is a valuable tool in the deception philosophy. It should be a tool to be considered when researching additional functionality or features for your IDS infrastructure.

Using deception technologies and philosophies is a great approach when being used as a layer in an IDS program. But should be considered by organizations that have a highly mature information security program that already has a good grasp of what is typical behavior on their network. This knowledge helps save time in installing and configuring any deception technologies that are out there. Like with any security tool, there is no such thing as silver bullet that will catch or stop everything. Using a defense in depth approach is the best recommendation for any security program. The goal is to make as many speed bumps as possible to slow down attackers. Speed bumps give security professionals time to discover the presence and then eradicate them from their environment. Deception tools can help with this approach and work as

part of a larger IDS methodology to gain useful insight into activity on the network and where a security analyst needs to be looking.

References

Attivo Networks At-a-Glance. (2016). Retrieved from Attivo Networks website: https://attivonetworks.com/documentation/Attivo_Networks-At-a-Glance.pdf

Attivo Networks. (2015, October 5). *Attivo Networks Art of Deception* [YouTube]. Retrieved from https://www.youtube.com/watch?v=Dx1SQqcaXho&feature=youtu.be

Beyer, R. (Producer). (2013). *The ghost army* [DVD]. USA: PBS.

Intrusion detection system. (n.d.). In *Wikipedia, the free encyclopedia*. Retrieved June 10, 2016, from https://en.wikipedia.org/wiki/Intrusion_detection_system

Know what is lurking in Your Network. (2015). Retrieved from Attivo Networks website: https://attivonetworks.com/documentation/Attivo_Networks-Understanding_Active_Deception.pdf

Lateral Movement: Beyond Initial Exploitation. (2015). Retrieved from Attivo Networks website: https://attivonetworks.com/documentation/Attivo_Networks-Lateral_Movement.pdf

Leveraging Deception for Visibility into Inside-the-Network Threats. (2016). Retrieved from Attivo Networks website: https://attivonetworks.com/documentation/Attivo_Networks-Deception_Visibility.pdf

Richardson, R. (2015). *Security startups tackle the art of deception techniques*. Retrieved from TechTarget website: http://searchsecurity.techtarget.com/opinion/Security-startups-tackle-the-art-of-deception-techniques

Russinovich, M. (2014, May 2). PsExec. Retrieved from https://technet.microsoft.com/en-us/sysinternals/psexec.aspx

"Active Deception" - Luring Cybercriminals to Reveal Themselves. (2015). Retrieved from Attivo Networks website: https://attivonetworks.com/documentation/Attivo_Networks-Active-Deception_White-Paper.pdf